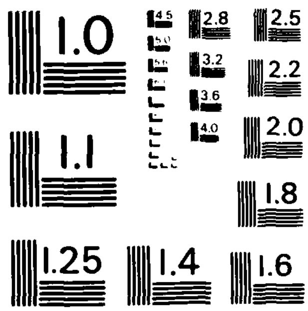
AD-A131 470 REPRESENTATION OF INFORMATION IN SOFTWARE DOCUMENTATION (U) GENERAL ELECTRIC CO ARLINGTON VA DATA AND INFORMATION SYSTEMS D A BOEHM-DAVIS JUL 83 1/1

UNCLASSIFIED GEC/DIS/TR-83-388200-8 N00014-79-C-0595 F/G 9/2 NL

END
DATE FILMED
9-83
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A131470

GEC/DIS/TR-83-388200-8

REPRESENTATION OF INFORMATION IN SOFTWARE DOCUMENTATION

Deborah A. Boehm-Davis

Software Management Research and Ada Development
Data and Information Systems
General Electric Company
1755 Jefferson Davis Highway
Arlington, Virginia 22202

DTIC
ELECTE
AUG 17 1983
S D

July 1983

DTIC FILE COPY

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

83 08 15 009

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER TR-83-388200-8	2. GOVT ACCESSION NO. AD-A131470	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Representation of information in software documentation		5. TYPE OF REPORT & PERIOD COVERED Final Report: 1 Aug 79 to 15 Jul 83 6. PERFORMING ORG. REPORT NUMBER GEC/DIS/TR-83-388200-8
7. AUTHOR(s) Deborah A. Boehm-Davis		8. CONTRACT OR GRANT NUMBER(s) N00014-79-C-0595
9. PERFORMING ORGANIZATION NAME AND ADDRESS Data & Information Systems General Electric Company 1755 Jefferson Davis Hwy., Arlington VA 22202		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61153N 42; RR04209 RR0420901; NR 196-160
11. CONTROLLING OFFICE NAME AND ADDRESS Engineering Psychology Group, Code 442 Office of Naval Research Arlington, VA 22217		12. REPORT DATE July 1983 13. NUMBER OF PAGES 20
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

Technical monitor: Dr. John J. O'Hare

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Software engineering, Software documentation, Software experiments, Modern programming practices, Software human factors, Program Design Languages (PDLs), Flowcharts, Petri nets, Resource Diagrams

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a series of six experiments designed to evaluate the performance of programmers using different forms of software documentation. The results from the first five experiments suggested that for sequential programs, succinct symbology leads to better performance than English prose, and a branching spatial arrangement provides the clearest display of control flow. The final experiment suggested that the most appropriate type of documentation for concurrent programs is different than

the most appropriate type of documentation for strictly sequential programs. For simple programs and simple modifications, it was not crucial whether interprocess communications or control-flow information was highlighted in the documentation format. For more complex problems, it appeared that control-flow information was not necessary, and, in fact, may have interfered with performance.

This report also describes the contents of the book Human Factors in Computer Systems. This book is being published by Ablex Publishers and it contains several papers selected from the conference held in Gaithersburg, Maryland in March 1982.

GEC/DIS/TR-83-388200-8

REPRESENTATION OF INFORMATION IN SOFTWARE DOCUMENTATION

Deborah A. Boehm-Davis

Software Management Research and Ada Development
Data and Information Systems
General Electric Company
1755 Jefferson Davis Highway
Arlington, Virginia 22202

Accession For
NTIS GRA&I X
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist Avail and/or Special
C

Submitted to:

Office of Naval Research
Engineering Psychology Group
Arlington, Virginia

Contract: N00014-79-C-0595
Work Unit: NR 196-160

July 1983

Approved for public release; distribution unlimited. Reproduction in whole or in part is permitted for any purpose of the United States Government.

TABLE OF CONTENTS

INTRODUCTION

For the past four years (1 AUG 79 - 15 JUL 83), we have been examining the nature of software documentation. The efficiency with which programming tasks are performed is determined in part by how thoroughly a programmer understands the design or function of the system under consideration. The thoroughness of a programmer's understanding, in turn, depends heavily on the quality of the documentation available. The research we conducted was therefore involved in identifying and validating human engineering principles to improve the ability of documentation to assist programmers in understanding programming systems.

Our approach to evaluating different forms of documentation was to investigate how various characteristics of presentation affect the performance of programmers on typical software-related tasks. There are two primary dimensions for categorizing how available documentation aids configure the information they present to programmers: the type of symbology in which information is presented, and the spatial arrangement of that information. The interrelation of these two dimensions describes generic types of documentation not necessarily embodied in existing techniques. The symbology dimension includes narrative text, constrained language, and ideograms. The spatial arrangement of information dimension was represented by sequential, branching, and hierarchical arrangements.

EXPERIMENTS

In this research program, we completed four experiments to investigate the effects of the type of symbology and the spatial arrangement. In the first experiment (Tech. Rep. 80-388200-2), 72 professional programmers were presented with documentation for each of three modular-sized computer programs. The participants answered a series of comprehension questions for each program using only the documentation (i.e., they were not given the actual program listing). The questions were presented interactively on a CRT and consisted of three different types. For forward-tracing questions, the participants were given the values for a set of conditions in the program. Their task was to trace through the documentation and find the first statement executed under those conditions. For backward-tracing questions, they were required to locate a given statement within the documentation and then determine the set of conditions which led to that point. For the input-output questions, they were given input data and were asked to determine the value of particular variables at a later point in the program.

Both forward and backward-tracing questions were answered more quickly from documentation presented in PDL or ideograms than in normal English. On the average, forward-tracing questions were answered most quickly from a branching arrangement and backward-tracing questions were answered more quickly from the branching and hierarchical arrangements. An examination of the individual formats revealed that the sequential PDL, the branching PDL, and the branching ideogram versions were associated with very quick responses for both types of questions. For the input-output questions, no significant differences were found as a function of the type of symbology or the spatial arrangement. At the conclusion of the experimental session, participants were asked to list the type of symbology and the spatial arrangement they most preferred. PDL was the most preferred symbology and the branching spatial arrangement was the most preferred arrangement.

In the second experiment (Tech. Rep. 81-388200-3), 36 professional programmers were presented with documentation and partially completed code for the same three programs. The participants constructed a major section of code at the middle of each program. About fifteen lines were missing from the code. This section included the most complex decision structures present in the program.

Substantial differences in performance were associated with the type of symbology. Coding from the normal English formats took considerably longer (29.7 minutes) than coding from the PDL (20.5

minutes) or ideogram (23.9 minutes) versions. An examination of the error data showed a similar pattern: the normal English formats resulted in a mean of 2.4 errors, the PDL resulted in 0.8 error and the ideograms resulted in 1.4 errors.

The effect of spatial arrangement was not as great as the effect of symbology. Although not statistically significant, the branching arrangement appeared to be superior to the sequential and hierarchical arrangements, particularly in minimizing errors related to the control flow. A comparison of the individual formats revealed that the sequential PDL and the branching PDL resulted in the highest level of performance. The branching ideograms and the hierarchical ideograms were also associated with good performance. Of the nine formats, the sequential normal English version resulted in the lowest level of performance.

The participants' preferences for symbology and spatial arrangement were consistent with the time and error data. PDL was the symbology preferred most often and branching was the most preferred spatial arrangement.

In the third experiment (Tech. Rep. 81-388200-4), 36 professional programmers were asked to compare error-seeded program code to the same documentation formats in order to detect and correct the errors. There were three errors per program. These errors were selected from among those made during the coding task in Experiment 2. The participants were told that the errors were located in the center section of the programs but they were not told how many errors occurred in each program. The dependent variable was time to debug.

Again, substantial differences in performance were associated with the type of symbology. Debugging from normal English took longer (18.7 minutes) than debugging from either PDL (14.5 minutes) or ideograms (14.2 minutes). The overall effect of spatial arrangement was not pronounced. A comparison of the individual formats revealed that the sequential and branching PDL again led to a high level of performance as did the branching and hierarchical ideograms. The sequential normal English again resulted in very poor performance.

The participants had no preferences for the type of symbology but did prefer the branching spatial arrangement to the sequential and hierarchical arrangements.

In the first three experiments, normal English resulted in substantially longer response times than the other two symbologies. It appeared likely that at least part of this difference was due to the manner in which variable names were expressed. The normal English contained an English description of each variable while the PDL and ideograms contained the variables as they were used in the FORTRAN code. Thus, the normal English required more translation from the documentation to the code.

In the fourth experiment (Tech. Rep. 81-388200-5), an abbreviated English was substituted for the ideograms in order to assess the extent to which the variable names account for the symbology effect. The abbreviated English was identical to the normal English with the exception that the variable names were used rather than normal English descriptions. Thus, the abbreviated English was more succinct than the natural language but less succinct than the PDL.

The task in this experiment was to modify the three programs. The modifications required a minimum of three to five lines of additional code. Performance was measured by the time to code and debug the modifications and by the number of errors.

Although the effect of type of symbology was not pronounced in this experiment, the results reflected the trend that appeared quite strongly in the previous three experiments. The more succinct symbology, the PDL, was associated with better performance than the more verbose symbology, the normal English. The effect of spatial arrangement was quite strong in this experiment. The branching arrangement was considerably better for the modification task than the other two arrangements.

The participants' preferences for type of symbology and spatial arrangement in this experiment are consistent with preferences from the other experiments. PDL was the preferred symbology and branching was the preferred spatial arrangement.

The first four experiments in this series, then, produced slightly different results, depending on the type of experimental task: answering questions, coding, debugging, or modifying programs. That is, no one particular combination of symbology and spatial arrangement proved superior for all tasks. However, there was a sufficient degree of consistency to allow the formulation of two general principles to characterize the overall effects of symbology and spatial arrangement:

(1) the more succinct the symbology, the better the performance, and

(2) the branching arrangement provides the clearest display of control flow. (An example of the PDL branching documentation can be seen in Figure 1.)

The results from these experiments suggested that PDLs are the optimal documentation format for coding. The question that arose was why this form of documentation was superior to the other formats tested. The most probable explanation of this superiority was that PDL was the most code-like of all documentation formats tested. As a result, there is less translation required in mapping between the documentation and the code. If the amount of translation is a critical underlying factor, no single form of PDL will be optimal for all implementation languages. Rather, the optimal PDL will be one that is tailored toward the particular implementation language.

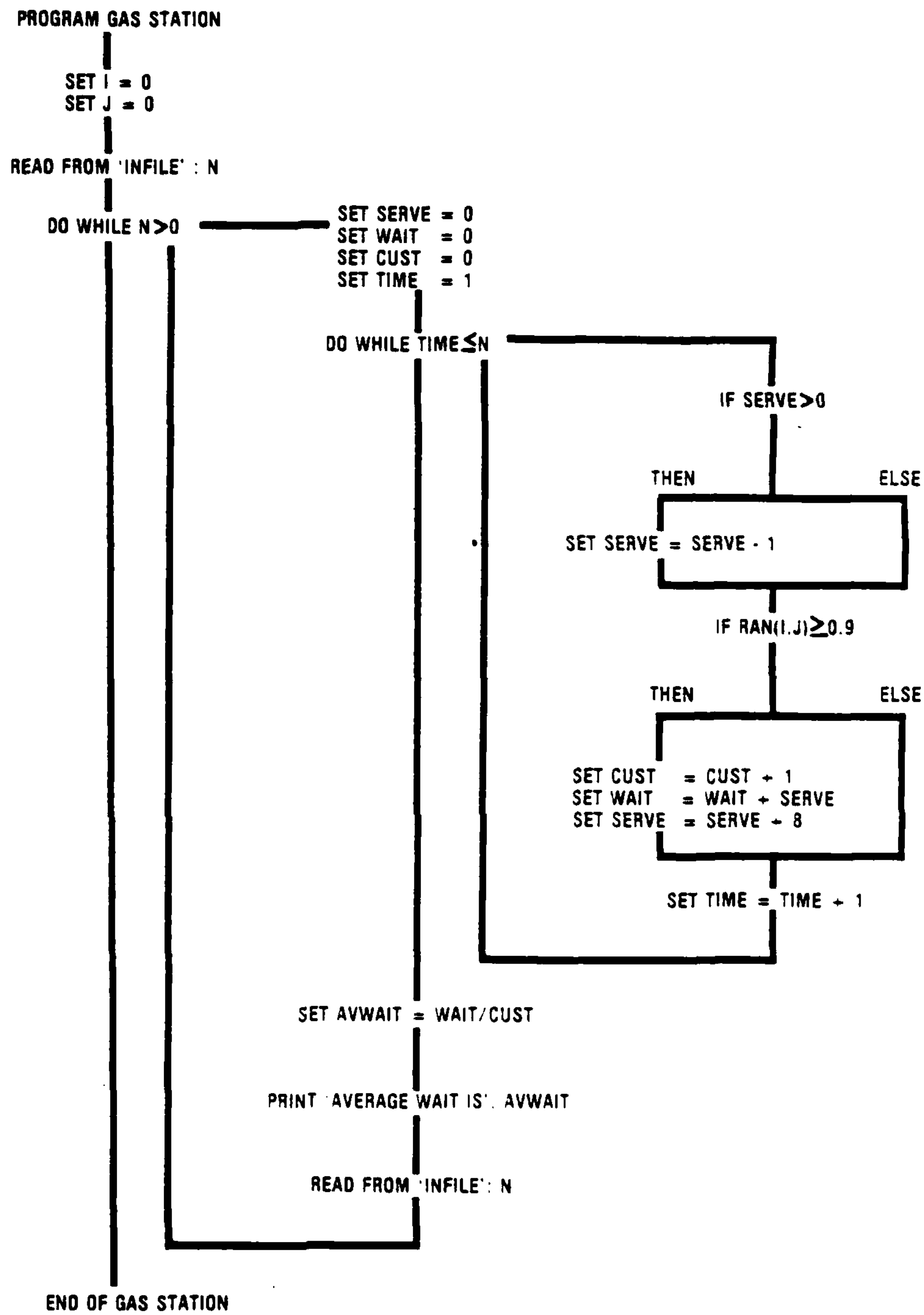


Figure 1. PDL Branching

A fifth experiment (Tech. Rep. 82-388200-6) was therefore conducted to determine the effectiveness of using a PDL specifically designed to aid in coding the corresponding implementation language. This was done by designing PDLs which reflected the syntax and features of several languages and examining the performance of programmers coding from these various PDLs in one of two implementation languages.

Twenty-four programmers were presented with three programs from which several lines had been deleted. Their task was to complete the code for each program in either FORTRAN or MACRO-11 (PDP-11 assembly language). Performance was measured by the time to code and debug the missing segment of code and by the number of errors.

The results from this experiment helped to shed light on our earlier finding of a substantial advantage for PDL over other formats. In this experiment, the fastest coding times occurred when there was a match between the PDL implementation language and the coding language, that is, using a MACRO-11 PDL when coding in MACRO-11 and using a FORTRAN PDL when coding in FORTRAN. This suggests that PDLs produced superior performance in our earlier studies since they required less translation in going from the design to the code than the other formats we tested.

The sixth and final experiment in this research program (Tech. Rep. 83-388200-7) extended the previous research from purely sequential programs into the domain of concurrent programming by examining performance on a modification task. The investigation of documentation for concurrent processing is especially important since this form of processing is used extensively in embedded computer systems which must monitor and control a number of hardware interfaces simultaneously. Embedded software now consumes over half of the Department of Defense software budget. Examples of embedded applications include systems for missile guidance, aircraft flight control, and multiplexing of communication channels.

In this experiment, new forms of documentation were constructed to allow for the representation of interprocess communications information, which does not exist in purely sequential programs. Seventy-two programmers were asked to make either a data-structure or control-flow modification to each of three programs.

Substantial differences in completion time were observed among the three types of documentation formats. For both kinds of modification (control flow or data structure), the resource diagrams led to the best performance while Petri nets led to the poorest performance. Two things should be noted, though. First, the data suggest that the differences among documentation formats are not very pronounced for all cases; the text search program provided the most striking differences. Second, the modifications used in this experiment were simple and did not require many control-flow changes; this will not always be the case with modifications.

The participants' choices for the easiest to use documentation format and their previous familiarity with one of the documentation formats lead to an interesting observation. Although, overall, 68% of the programmers had used PDLs before this experiment and 71% of them chose it as the easiest to use, the time required to make the modifications with the PDLs was in between the other documentation formats, for the two types of task modification.

Taken as a whole, the data suggest that the most appropriate type of documentation for concurrent processing (resource diagram) is different than the most appropriate type of documentation for strictly sequential processing (PDL). For modifications to concurrent processing programs, at least for simple programs and simple modifications, it is not crucial whether interprocess communications or control-flow information is highlighted in the documentation format. For more complex problems, it would appear that control-flow information is not necessary, and, in fact, may interfere with making the modification. In addition, the results suggest that data-structure modifications are easier than control-flow changes.

CONCLUSIONS

Overall, the work we have done has led us to several important observations about the nature of documentation. It has shown us that the communication of control-flow information is essential in documentation for sequential programs. Given that PDLs show this type of information clearly in a way which is familiar to programmers, the research suggests that PDLs should be used more often as documentation. In fact, we have received favorable feedback from several Navy and non-military installations who have begun to use PDLs as a result of our research. Our work has also been cited by other researchers in the field, such as Atwood, Ramsey, and Van Doren, and Shneiderman.

The research further suggests that we need to be careful in generalizing our results. The final experiment suggested that the ideal form of documentation may be dependent upon the type of program being used. In this experiment, which used concurrent programs, the representation of interprocess communications information was more important than the representation of control-flow information in determining performance on a modification task.

A collateral activity conducted under this contract was the planning and production support for the Human Factors in Computer Systems Conference and its proceedings, which was held in Gaithersburg, Maryland in March 1982. The conference was a rousing success, attracting over 900 people from all over the United States and Europe. The proceedings were distributed at the conference and many more have been sent out in response to requests. In fact, the number of requests required several additional printings of the proceedings. In addition, a book of selected papers from the conference will be published in 1983 by Ablex Press, with John Thomas and Mike Schneider as co-editors. This book will include the following chapters.

Carroll, J. & Mack, R. Learning to use a word processor: By doing, by thinking, and by knowing.

This paper describes studies of learning to use a commercial word processer. Instructional materials which were designed to encourage active learning were found to produce performance which was superior to instructional materials which only encouraged passive learning. Implications for further development of training materials is discussed.

Ehrlich, K. & Soloway, E. An empirical investigation of the tacit plan knowledge in programming.

In this paper, Ehrlich and Soloway describe an experimental technique where programmers are asked to "fill in the blank" in a program from which some number of lines has been deleted. The responses made to the blank lines were used to infer what kind of knowledge is used by expert programmers. On the basis of the responses they received using a simple Pascal problem, the authors propose that expert programmers have acquired information which is chunked into "plan knowledge."

Furnas, G., Landauer, T., Gomez, L., & Dumais, S. Statistical semantics: Analysis of the potential performance of keywork information systems.

Furnas, Landauer, Gomez, and Dumais describe the results of several studies which were conducted to assess peoples' abilities to choose descriptions or names for objects so that someone else could identify that object from a set of alternatives. Their results suggest that people use a large variety of terms to describe even very common objects. The implications of this result for the viability or choice of keywords in menu systems is discussed.

Goldsmith, T. & Schvaneveldt, R. The role of integral information displays in decision making.

This paper describes a series of four experiments which evaluate the usefulness of integrating information from several sources into a holistic display. Using various types of integral displays, Goldsmith and Schvaneveldt found that the integral displays generally led to better performance than separable displays. The implications for the display of information for decision-making purposes is discussed.

Malone, T. Heuristics for designing enjoyable user interfaces: Lessons from computer games.

Malone describes challenge, fantasy, and curiosity as three features which make computer games fun. In this paper, he uses these features (derived from earlier empirical studies) to describe a set of heuristics for designing enjoyable user interfaces. These guidelines are designed to be applied to both "toys" and "tools", two distinct categories of computing systems.

McNicholl, D. & Magel, K. The subjective nature of programming complexity.

This paper describes sets of measurements which were made in an attempt to characterize the complexity of programs produced by college students. The results suggest that complexity is a highly subjective trait; students did not agree on complexity rankings for their programs. While the overall ratings suggest that program size is the best predictor of complexity, individuals' rankings were often better explained by measures of the effort required to produce the programs or by complexity rankings computed on the program specifications.

Reisner, P. Formal grammer as a tool for building usable computer systems.

This paper describes an attempt to use formal grammar as a design tool, using ROBART, an IBM interactive graphics system, as an example. The processes of incorporating "cognitive information" into the grammar and predicting performance in using the system are described. Tests of the ROBART system using the prediction assumptions are described.

Savage, R., Habineck, J., & Barnhart, T. The design, simulation, and evaluation of a menu-driven user interface.

This paper describes the design and evaluation of a menu-driven user interface. A group of participants was asked to perform a set of tasks using the interface that had been designed for the experiment. On the basis of protocols which showed what errors were made in using the system, the interface was modified and a new group of participants used the interface. The results from this phase suggested that performance, as measured by number of errors, was improved by the modifications to the interface. On the basis of this research, some guidelines for menu design are discussed.

Sheppard, S., Kruesi, E., & Bailey, J. An empirical evaluation of software documentation formats.

Sheppard, Kruesi, and Bailey describe the results of three experiments designed to examine the effects of documentation formats on the performance of programmers on different software-related tasks. Performance of coding, debugging, and modification tasks was measured by speed and accuracy measures. The results suggest that succinct symbology should be used in documentation rather than English prose. Further, presenting the information in a branching arrangement seemed to provide for the lowest overall performance times and the fewest number of errors.

SCIENTIFIC PERSONNEL WHO WORKED ON PROGRAM

John Bailey
Deborah Boehm-Davis
Paul Chase
Bill Curtis
Andrew Fregly
Betsy Kruesi
Pete McEvoy
Dave Morris
Sylvia Sheppard
Bryan Wolfe

TECHNICAL REPORTS

Curtis, B. Measurement and experimentation in software engineering (Tech. Rep. 80-388200-1). General Electric Co., April 1980. (AD-A085827)

Sheppard, S. B., Kruesi, E., & Curtis, B., The effects of symbology and spatial arrangement on the comprehension of software specifications (Tech. Rep. 80-388200-2). General Electric Co., August 1980. (AD-A091614)

Sheppard, S. B., & Kruesi, E. The effects of the symbology and spatial arrangement of software specifications in a coding task (Tech. Rep. 81-388200-3). General Electric Co., February 1981. (AD-A097029)

Sheppard, S. B., Bailey, J. W., & Kruesi, E. The effects of the symbology and spatial arrangement of software specifications in a debugging task (Tech. Rep. 81-388200-4). General Electric Co., August 1981. (AD-A104541)

Sheppard, S. B., Bailey, J. W., & Kruesi, E. The effects of the symbology and spatial arrangement of software documentation in a modification task (Tech. Rep. 81-388200-5). General Electric Co., December 1981. (AD-A109833)

Boehm-Davis, D. A., Sheppard, S. B., Bailey, J. W., & Kruesi, E. An empirical evaluation of language-tailored PDLs (Tech. Rep. 82-388200-6). General Electric Co., May 1982. (AD-A115632)

Boehm-Davis, D. A., & Fregly, A. M., Documentation of concurrent programs (Tech. Rep. 83-388200-7). General Electric Co., July 1983.

Boehm-Davis, D. A. Representation of information in software documentation. (Tech. Rep. 83-388200-8). General Electric Co., July 1983.

ARCHIVAL PUBLICATIONS

Boehm-Davis, D. A. On the use of psychology in evaluating software documentation. Behavior Research Methods and Instrumentation. 1983, 15, 217-221.

Boehm-Davis, D. A., & Fregly, A. M. Documentation of concurrent programs. In Proceedings of the CHI '83 Conference, Boston, MA, December 1983, in press.

Boehm-Davis, D. A., Sheppard, S. B., & Bailey, J. W. An empirical evaluation of language-tailored PDLs. In Proceedings of the Twenty-First Annual Technical Symposium on Computing and Government: Interactions and Achievement. Washington, D. C.: Association for Computing Machinery, 1982.

Boehm-Davis, D. A., Sheppard, S. B., & Bailey, J. W. An empirical evaluation of language-tailored PDLs. In Proceedings of the 26th Annual Meeting of the Human Factors Society. Santa Monica, CA: The Human Factors Society, 1982, 984-988.

Curtis, B., In search of software complexity. In Workshop on Quantitative Software Models for Reliability, Complexity, and Cost, 1980, 95-106.

Curtis, B. Measurement and experimentation in software engineering. In Proceedings of the IEEE, September 1980.

Curtis, B., Program complexity and software errors: A front end for reliability. In Proceedings of the Fourth Software Engineering Workshop, 1979, 217-238.

Curtis, B., Experimental evaluation of software characteristics. In A. Perlis, F. Sayward, & M. Shaw (Eds.), Software Metrics: An analysis and evaluation. Cambridge, MA: MIT Press, 1981.

Curtis, B., The measurement of software quality and complexity. In A. Perlis, F. Sayward, & M. Shaw (Eds.), Software Metrics: An analysis and evaluation. Cambridge, MA: MIT Press, 1981.

Sheppard, S., Bailey, J., & Kruesi, E. An empirical evaluation of software documentation formats. In J. Thomas, & M. Schneider (Eds.), Human Factors in Computer Systems. Norwood, NJ: Ablex Publishing Corp., in press.

Sheppard, S. B., & Kruesi, E., The effects of the symbology and spatial arrangement of software specifications in a coding task. In Proceedings Trends and Applications 1981: Advances in Software Technology, 1981, 7 - 13.

Sheppard, S. B., Kruesi, E., & Bailey, J. W. An empirical evaluation of software documentation formats. In Proceedings of Human Factors in Computer Systems. Washington, D. C.: Association for Computing Machinery, 1982, 121-124.

Sheppard, S. B., Kruesi, E., & Curtis, B., The effects of symbology and spatial arrangement on comprehending software specifications. In Proceedings of the Fifth International Conference on Software Engineering, 1981, 207-214.

Thomas, J. & Schneider, M. (Eds.). Human Factors in Computer Systems. Norwood, NJ: Ablex Publishing Corp., in press.

TECHNICAL REPORTS DISTRIBUTION LIST

OFFICE OF NAVAL RESEARCH

Engineering Psychology Group

TECHNICAL REPORTS DISTRIBUTION LIST

OSD

CAPT Paul R. Chatelier
Office of the Deputy Under
Secretary of Defense
OUSDRE (E&LS)
Pentagon, Room 3D129
Washington, DC 20301

Dr. Dennis Leedom
Office of the Deputy Under
Secretary of Defense (C^3I)
Pentagon
Washington, DC 20301

Department of the Navy

Engineering Psychology Group
Office of Naval Research
Code 442 EP
Arlington, VA 22217 (2 cys.)

Aviation & Aerospace Technology
Programs
Code 210
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Communication & Computer
Technology Programs
Code 240
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Information Sciences Division
Code 433
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Dr. J. S. Lawson
Naval Electronic Systems Command
NELEX-06T
Washington, DC 20360

Department of the Navy

Tactical Development & Evaluation
Support Programs
Code 230
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Manpower, Personnel & Training
Programs
Code 270
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Special Assistant for Marine Corps
Matters
Code 100M
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

CDR James Offutt, Officer-in-Charge
ONR Detachment
1030 East Green Street
Pasadena, CA 91106

Director
Naval Research Laboratory
Technical Information Division
Code 2627
Washington, DC 20375

Dr. Michael Melich
Communications Sciences Division
Code 7500
Naval Research Laboratory
Washington, DC 20375

Dr. Robert E. Conley
Office of Chief of Naval Operations
Command and Control
OP-094H
Washington, DC 20350

Department of the Navy

Dr. Robert G. Smith
Office of the Chief of Naval Operations, OP987H
Personnel Logistics Plans
Washington, DC 20350

Dr. Alfred F. Smode
Training Analysis and Evaluation Group
Orlando, FL 32813

Dr. Gary Poock
Operations Research Department
Naval Postgraduate School
Monterey, CA 93940

Dean of Research Administration
Naval Postgraduate School
Monterey, CA 93940

Dr. L. Chmura
Naval Research Laboratory
Code 7592
Computer Sciences & Systems
Washington, DC 20375

Chief, C3 Division
Development Center
MCDEC
Quantico, VA 22134

Commander
Naval Air Systems Command
Human Factors Programs
NAVAIR 334A
Washington, DC 20361

Commander
Naval Air Systems Command
Crew Station Design
NAVAIR 5313
Washington, DC 20361

Commander
Naval Electronics Systems Command
Human Factors Engineering Branch
Code 81323
Washington, DC 20360

Dr. George Moeller
Human Factors Engineering Branch
Submarine Medical Research Lab
Naval Submarine Base
Groton, CT 06340

Department of the Navy

Combat Control Systems Department
Code 35
Naval Underwater Systems Center
Newport, RI 02840

Human Factors Department
Code N-71
Naval Training Equipment Center
Orlando, FL 32813

CDR Norman E. Lane
Code N-7A
Naval Training Equipment Center
Orlando, FL 32813

Dr. A.L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, DC 20380

HQS, U. S. Marine Corps
ATTN: CCA40 (Major Pennell)
Washington, DC 20380

Commanding Officer
MCTSSA
Marine Corps Base
Camp Pendleton, CA 92055

Human Factors Technology Admin.
Office of Naval Technology
Code MAT 0722
800 North Quincy Street
Arlington, VA 22217

Mr. Lawrence Lindley
Naval Avionics Center
Code 321
6000 East 21st Street
Indianapolis, IN 46218

Mr. Philip Andrews
Naval Sea Systems Command
NAVSEA 03416
Washington, DC 20362

Larry Olmstead
Naval Surface Weapons Center
NSWC/DL
Code N-32
Dahlgren, VA 22448

Department of the Navy

Mr. Ronald Leask
Naval Underwater Systems Center
Code 3251
Smith Street
New London, CT 06320

Navy Personnel Research and
Development Center
Planning & Appraisal Division
San Diego, CA 92152

Mr. Stephen Merriman
Human Factors Engineering Div.
Naval Air Development Center
Warminster, PA 18974

Mr. Jeffrey Grossman
Human Factors Branch
Code 3152
Naval Weapons Center
China Lake, CA 93555

Dean of Academic Departments
U. S. Naval Academy
Annapolis, MD 21402

Dr. S. Schiflett
Human Factors Section
Systems Engineering Test
Directorate
U.S. Naval Air Test Center
Patuxent River, MD 20670

CDR C. Hutchins
Code 55
Naval Postgraduate School
Monterey, CA 93940

Office of the Chief of Naval
Operations (OP-115)
ATTN: Dr. Robert Carroll
Washington, DC 20350

Mr. Marshall R. Potter
Project Management Support Branch
System Effectiveness & Component
Engineering Division
Code NAVELEX 8143
Washington, DC 20360

Department of the Navy

Commanding Officer
Naval Health Research Center
San Diego, CA 92135

Commander, Naval Air Force,
U. S. Pacific Fleet
ATTN: Dr. James McGrath
Naval Air Station, North Island
San Diego, CA 92135

Dr. Robert Blanchard
Navy Personnel Research and
Development Center
Command and Support Systems
San Diego, CA 92152

Human Factors Engineering Branch
Code 1226
Pacific Missile Test Center
Point Mugu, CA 93042

Mr. John Impagliazzo
Code 101
Naval Underwater Systems Center
Newport, RI 02840

Mr. Harry Crisp
Code N-51
Combat Systems Department
Naval Surface Weapons Center
Dahlgren, VA 22448

Department of the Army

Mr. J. Barber
HQS, Department of the Army
DAPE-MBR
Washington, DC 20310

Dr. Edgar M. Johnson
Technical Director
U. S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Director, Organizations and
Systems Research Laboratory
U. S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Department of the Army

Technical Director
U.S. Army Human Engineering Labs
Aberdeen Proving Ground, MD 21005

Department of the Air Force

U. S. Air Force Office of Scientific Research
Life Sciences Directorate, NL
Bolling Air Force Base
Washington, DC 20332

AFHRL/LRS TDC
ATTN: Susan Ewing
Wright-Patterson AFB, OH 45433

Chief, Systems Engineering Branch
Human Engineering Division
USAF AMRL/HES
Wright-Patterson AFB, OH 45433

Dr. Earl Alluisi
Chief Scientist
AFHRL/CCN
Brooks AFB, TX 78235

Foreign Addresses

Director, Human Factors Wing
Defence & Civil Institute of Environmental Medicine
Post Office Box 2000
Downsview, Ontario M3M 3B9
CANADA

Other Government Agencies

Defense Technical Information Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 (12 cys.)

Dr. Clint Kelly
Director, System Sciences Office
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

Other Government Agencies

Dr. M. Montemerlo
Human Factors & Simulation Technology, RTE-6
NASA HQS
Washington, DC 20546

Other Organizations

Dr. Jesse Orlansky
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

Dr. Robert T. Hennessy
NAS - National Research Council (COHF)
2101 Constitution Avenue, N.W.
Washington, DC 20418

Dr. Robert C. Williges
Department of Industrial Engineering and OR
Virginia Polytechnic Institute and State University
130 Whittemore Halle
Blacksburg, VA 24061

Mr. Edward M. Connelly
Performance Measurement Associates, Inc.
410 Pine Street, S.E.
Suite 300
Vienna, VA 22180

Dr. J. O. Chinnis
Decision Science Consortium
Suite 721
7700 Leesburg Pike
Falls Church, VA 22043

Dr. Richard Pew
Bolt, Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02238

Psychological Documents (3 cys.)
ATTN: Dr. J. G. Darley
N-565 Elliott Hall
University of Minnesota
Minneapolis, MN 55455

LME
8